FROST DEPTH PREDICTION

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Meng Luo

In Partial Fulfillment
for the Degree of
MASTER OF SCIENCE

Major Department:
Statistics

July 2014

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Frost Depth Prediction

By

Meng Luo

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Gang Sheng
Chair

Dr. Rhonda Magel

Dr. Seung Won Hyun

Dr. Xinhua Jia

Approved:

6/30/2014
Date

Dr. Rhonda Magel
Department Chair

ABSTRACT

The purpose of this research project is to develop a model that is able to accurately predict frost depth on a particular date, using available information. Frost depth prediction is useful in many applications in several domains. For example in agriculture, knowing frost depth early is crucial for farmers to determine when and how deep they should plant. In this study, data is collected primarily from NDAWN (North Dakota Agricultural Weather Network) Fargo station for historical soil depth temperature and weather information. Lasso regression is used to model the frost depth. Since soil temperature is clearly seasonal, meaning there should be an obvious correlation between temperature and different days, our model can handle residual correlations that are generated not only from time domain, but space domain, since temperatures of different levels should also be correlated. Furthermore, root mean square error (RMSE) is used to evaluate goodness-of-fit of the model.

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Dr. Gang Shen, for his knowledgeable guidance, patience and encouragement, and for giving me the research directions. Without his support, this thesis would not be possible.

Secondly, I would like to thank Dr. Xinhua Jia for her generous help on my study. I would also like to extend thanks to all my other committee members, Dr. Rhonda Magel and Dr. Seung Won Hyun, for their time and for their valuable suggestions on my thesis. Furthermore, I would like to thank all faculty members in the Department of Statistics at NDSU for their kind help during my course studies.

I also would like to thank Mr. Jade Sandbulte for his help on correcting my English writting.

Thirdly, I would like express my gratitude to my parents and brother, for their unconditional support and encouragement.

At the end, special thanks to my husband, Jianfei Wu, for his encouragement, love, and understanding that enables this thesis to be completed. I am also very proud of my daughters, Luoxi Wu and Rachel Wu, for their loveliness and intelligence. Your support make everything possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

Most soils at high altitudes or elevations are seasonally frozen soil. Land freeze-thaw is a seasonal transition process where the soil temperature drops below 0°C then rise above 0°C.

Predicting the depth to which soils may freeze and thaw can be helpful for guiding city plan. Freezing and thawing can cause land geological disasters. Soil freezing produces volume expansion, and melting of the soft soil caused subsidence. It often cause building foundation damage; subsidence of ground; in slope area lead to landslide and collapse ;road subgrade deformation, and threat to traffic safety, transport etc.

Another reason for investigating predictive modes for frost depth is for environmental phenomena, such as runoff and flooding associated with rainfall and snowmelt on frozen soil. In the Fargo-Moorhead area, due to the presence of ice, snow melt over frost soil can lead to the reduction in soil infiltration capacity and an increase in spring stream flow to Red River which probably result in a greater potential risk for flood.

Furthermore, in agriculture world, frost depth predicting is important for its use in managing agricultural activities and water resources. Knowing frost depth early is crucial for farmers to determine when and how deep they should plant. More frequent freeze thaw cycles (FTCs) may affect ecosystem diversity and productivity because freeze-thaw cycles cause changes in soil physical properties and affect water movement in the landscape. Soil freezing and thawing influence the infiltration of water and subsequent redistribution, runoff generation.

Frost depth prediction or similar studies have already attracted researchers from various domains, either from academy or from industry, for several decades. Many methods and approaches have been developed with the advances of statistic and computing technologies. Farrngton and Gildea [1] presented a frost penetration prediction model using numerical simulation, statistical regression, spatial interpolation, and GIS. Using their

methods, they concluded that seasonal maximum frost penetration depth can be reliably estimated by the relationship to the actual annual freezing degree index (AFDI), as long as a pavement-specific relationship is derived using meteorological data that account for region-specific weather dynamics. A regression of maximum seasonal frost penetration depth (derived from dynamic simulations of temperature and moisture flux in a pavement structure using actual climatic data) on AFDI show a strong positive correlation and was useful for fitting a linear equation to the median and 90% upper prediction limit of maximum frost penetration depth [1].

Thordarson [2] developed a model for road surface temperature and sub-base frost depth prognosis. The model is connected to a frost depth and sb-base temperature sensor and the Automatic Weather Station which enables accurate real-time operation of the model. Using input data from a 5 day weather forecast, the model is capable of accurately predicting the development of freeze or thaw in the road sub-base.[2]

Haithem and et al [3] introduced a simplified model to predict the frost penetration in Manitoba. The goal of their research is to provide better understanding to the seasonal variation of the properties of pavement materials. The climatic and seasonal monitoring data for the Oak Lake test section were used in the model. The experimental result was gree with the result from Northern Ontario frost penetration model.

Lee [4] provided a frost indicator with methylene blue solution method to measure the frost depth.

The purpose of my research project is to develop a model that is able to accurately predict frost depth on a particular date using available information. In this study, data pertaining to historical soil depth temperature and weather information is collected primarily from two NDAWN (North Dakota Agricultural Weather Network) Fargo stations. Lasso regression [5] will be used to model the frost depth. The Lasso regression technique is selected in this research primarily due to its capability of pruning unimportant covariates.

Since ground temperature is clearly seasonal, which means there should be an obvious correlation between temperature and different days, our model should be able to handle residual correlations that are generated not only from the time domain but also the spatial domain. Through a preliminary investigation, Gupta's research [6] "A note on the asymptotic distribution of Lasso estimator for correlated data" will be used in this project. Furthermore, root mean square error (RMSE) will be used to evaluate goodness-of-fit of the model.

The historical soil depth temperature data used in this research has temperature records at several different soil depths in each year. This characteristic of the data provides us two options to build models on predicting frost depth. One option is to build a model to predict soil temperature[1] at each depth, and then uses an interpolation method to calculate frost depth from the predicted soil temperature values at each depth. The second option is to calculate soil frost depth values as the response variable first, and then build a model directly to predict frost depth. For the first option, two methods, namely "Modeling Soil Temperature First at Each Depth" and "Modeling Soil Temperature First on All Depths", will be discussed shortly. "Modeling Soil Temperature First at Each Depth" method builds a model on each depth, while "Modeling Soil Temperature First on All Depths" builds a single model on the whole data set including all depths data. For the second option, method "Modeling Directly on Frost Depth" will be presented. Each of the three methods have their own advantages and disadvantages, which will be discussed in great detail shortly.

The thesis is organized as follows: In Chapter 2, the data sets that are used in this study will be discussed in detail. Several steps of the pre-processing of the data sets will be discussed in Chapter 3. In Chapter 4, Lasso regression [5] technique and how it applied to the data sets with correlated residues will be discussed in great detail. Three frost depth modeling techniques using Lasso regression will be presented in Chapter 5. Extensive

[1]The response variable is thus temperature.

experiments and comparison will be conducted in Chapter 6, which will also present the final combined approach. Finally, conclusions will be made in Chapter 7.

CHAPTER 2. DATA DESCRIPTION

In this study, data is collected primarily from NDAWN (North Dakota Agricultural Weather Network) Fargo station; one data set has historical soil depth temperature and the other has historical weather information.

The first data set is from the Fargo Station Deep Soil Temperatures [7] website. On the website, there are soil temperature data for different depths of soil, ranging from 1cm to 1170 cm. Table 2.1. shows an example of the soil temperature data.

Table 2.1. An example of soil temperature data

Sta	Year	Mo	Day	Jday	1cmC	5cmC	10cmC	20cmC	...	250cmC
$FARG$	1994	1	1	1	-18.9	-2.1	-1.8	-1.4	...	7.5
$FARG$	1994	1	2	2	-19.9	-2.2	-1.9	-1.5	...	7.4
$FARG$	1994	1	3	3	-17.2	-2.4	-2.0	-1.6	...	7.3
$FARG$	1994	1	4	4	-20.9	-2.5	-2.1	-1.7	...	7.3

In Table 2.1., column Sta indicates the weather station. Due to space limitation, Table 2.1. only shows temperatures for several soil depths. In the original data file, there is temperature data for 23 levels of depths in total: 1 cm, 5 cm, 10 cm, 20 cm, 30 cm, 40 cm, 50 cm, 60 cm, 80 cm, 100 cm, 125 cm, 150 cm, 175 cm, 200 cm, and 250 cm. However, not all soil depths have temperature data for all years, thus further preprocessing is necessary on these data, which will be discussed in Chapter 3.

The second data set is Fargo_daily.csv data, which is from [7]. Table 2.2. shows an example of the data.

Table 2.2. An example of Fargo daily data

Sta	La	Lo	El	Year	Month	Day	Tmax	Tmin	...	Precip
$FARG$	46.897	-96.812	902	1994	1	1	-15.86	-21.54	...	M
$FARG$	46.897	-96.812	902	1994	1	2	-14.92	-26.61	...	M
$FARG$	46.897	-96.812	902	1994	1	3	-10.92	-24.05	...	M
$FARG$	46.897	-96.812	902	1994	1	4	-15.41	-28.04	...	M

In Table 2.2., Sta indicates station. La, Lo, and El are the abbreviation for latitude, longitude, and elevation, respectively. The second data set has the information of maximum temperature of the day (Tmax), the minimum temperature of the day (Tmin), the average temperature of the day (Tavg), the average bare soil temperature (Tbs), the average turf soil temperature (Tts), the average wind speed (WSavg), the maximum wind speed (WSmax), the average wind direction (WDavg), the total solar radiation (Solar), the total rainfall (Rainfall), the average dew point temperature (DP), the average wind chill temperature (WC), and precipitation information (Precip). Note that several of these columns are not shown in Table 2.2. due to space limitation.

CHAPTER 3. DATA PREPROCESSING

a) Preprocess Fargo Daily Data

In the Fargo Daily Data dataset, all cell values in Sta are identical as the values in La, Lo, and El as shown in Table 2.2.. We remove these columns first in the preprocessing step, since these columns will not have any predictive power in our regression models which will be discussed shortly.

Missing values in Rainfall are simply replaced with 0, since most values of this attribute are 0.

Table 3.1. An example of preprocessed Fargo daily data

Year	Month	Day	Tmax	Tmin	...	Precip
1994	1	1	-15.86	-21.54	...	M
1994	1	2	-14.92	-26.61	...	M
1994	1	3	-10.92	-24.05	...	M
1994	1	4	-15.41	-28.04	...	M

Table 3.1. shows the preprocessed data for the same data in Table 2.2..

b) Preprocess Soil Temperature Data

As mentioned above, there is a Soil Temperature Data file for each year, as shown in the example in Table 2.1.. The regression models, which will be discussed shortly, require every year to have the same sets of date. Thus the data of years 1993 and 2011 were removed first since they are not complete.

c) Merge data sets

For each soil depth of each date in the Soil Temperature Data dataset, we concatenate the data of Soil Temperature Data with the data from Fargo Daily Data of the same date. Take January 1. 1994 in Table 2.1. for example, there are in total 15 depth levels, i.e. 1cmC, 5cmC, ..., 250cmC, so after concatenation, there are 15 rows created, as Table 3.2. shows.

Table 3.2. An example of merged dataset

Year	Month	Day	Tmax	Tmin	...	Precip	AirTemp	Depth	Temperature
1994	1	1	-15.86	-21.54	...	M	-18.9	1	-2.1
1994	1	1	-15.86	-21.54	...	M	-18.9	5	-1.8
1994	1	1	-15.86	-21.54	...	M	-18.9	10	-1.4
1994	1	1	-15.86	-21.54	...	M	-18.9	20	-0.5
1994	1	1	-15.86	-21.54	...	M	-18.9	30	0.2
1994	1	1	-15.86	-21.54	...	M	-18.9	40	0.9
1994	1	1	-15.86	-21.54	...	M	-18.9	50	1.4
1994	1	1	-15.86	-21.54	...	M	-18.9	60	2
1994	1	1	-15.86	-21.54	...	M	-18.9	80	3
1994	1	1	-15.86	-21.54	...	M	-18.9	100	3.8
1994	1	1	-15.86	-21.54	...	M	-18.9	125	4.8
1994	1	1	-15.86	-21.54	...	M	-18.9	150	5.4
1994	1	1	-15.86	-21.54	...	M	-18.9	175	6.1
1994	1	1	-15.86	-21.54	...	M	-18.9	200	6.6
1994	1	1	-15.86	-21.54	...	M	-18.9	250	7.5

In this study, we want to predict the frost depth on each day (if there is any). As mentioned before, the frost depth is defined as the depth at which the soil temperature is 0°C and above which the soil temperature is greater than 0°C[1].

[1]There are some dates in the data that below the forst depth exists another sub-zero-temperature soil level. However this study only concentrate on the frost depth.

Figure. 3.1. shows the average soil temperature at each of the 15 depths for each date. From the figure we can see that there are only a fraction of dates that have a frost depth.

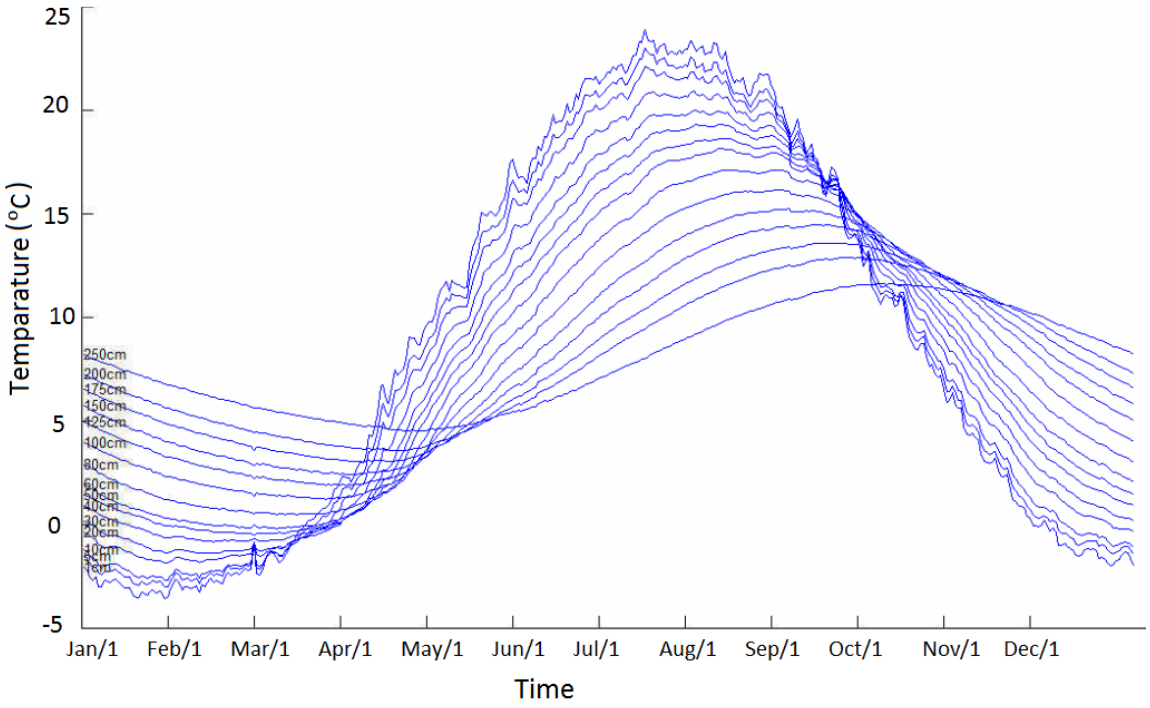

Figure 3.1. Average soil temperature at each of the 15 depths for each date.

Furthermore, the sets of dates for which there is a frost depth also differ from year to year as Figures. 3.2., 3.3., and 3.4. show. Figures. 3.2., 3.3., and 3.4. show the dates that have a frost depth from year 1993 to year 2010. The first two columns show the date of each year. The remaining columns indicate whether there is a frost depth (”1” indicates a frost depth, ”0” indicates otherwise) for each year. For example in Figure. 3.2., there is a frost depth in January 9. 2002 since the cell is ”1”, while there is no frost depth in January 9. 2001 since the cell is ”0”.

Month	Day	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
1	9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
2	10	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
2	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
2	18	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
2	19	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
2	21	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	22	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	23	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
2	24	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
2	25	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
2	26	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
2	27	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
2	28	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
2	29	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	6	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
3	7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
3	10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
3	11	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
3	12	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
3	13	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1
3	14	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1
3	15	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1

Figure 3.2. Dates that have frost depth.The first two columns show the date of each year. The remaining columns indicate whether there is a frost depth (”1” indicates have a frost depth, ”0” indicates otherwise) for each year.

Month	Day	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
3	16	0	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1
3	17	0	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1
3	18	0	1	1	1	0	1	1	0	0	0	1	0	0	0	1	1	0	1
3	19	0	1	1	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0
3	20	0	1	1	1	0	1	1	1	1	0	1	0	0	0	0	1	0	0
3	21	0	1	1	1	0	1	1	1	1	0	1	0	0	0	1	0	0	1
3	22	0	0	1	1	0	1	1	1	1	0	1	0	1	0	1	1	1	1
3	23	0	0	1	1	0	1	1	1	0	0	1	1	1	0	1	0	1	1
3	24	1	0	1	0	0	0	1	0	0	0	1	1	1	0	1	1	1	1
3	25	1	0	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	0
3	26	1	0	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	1
3	27	1	0	1	0	0	1	1	0	0	1	1	1	1	1	1	1	0	1
3	28	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	1
3	29	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	1
3	30	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	1
3	31	1	0	1	0	1	0	1	0	1	1	1	0	1	1	1	1	1	0
4	1	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	0	0
4	2	1	0	1	1	1	0	1	0	1	0	1	0	1	1	1	1	0	0
4	3	1	0	1	1	1	0	1	0	1	0	1	0	1	1	1	1	0	0
4	4	1	0	0	1	1	0	1	0	1	0	0	0	1	1	0	1	0	0
4	5	1	0	0	1	1	0	1	0	1	1	1	0	1	1	0	1	0	0
4	6	1	0	1	1	1	0	1	0	1	1	1	0	1	1	0	1	0	0
4	7	1	0	1	1	0	0	1	0	1	1	1	0	1	1	1	1	0	0
4	8	1	0	1	1	0	0	1	0	1	1	1	0	1	0	0	1	0	0
4	9	1	0	0	1	0	0	0	0	1	1	1	0	1	0	0	1	1	0
4	10	1	0	1	1	0	0	0	0	1	1	1	0	1	0	1	1	1	0
4	11	1	0	0	1	0	0	0	0	1	1	1	0	0	0	1	1	1	0
4	12	1	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	1	0
4	13	1	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	1	0
4	14	1	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	1	0
4	15	1	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0
4	16	1	0	1	1	1	0	0	0	1	1	1	0	0	0	1	0	1	0
4	17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
4	18	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	0	0	0
4	19	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0

Figure 3.3. Dates that have frost depth.The first two columns show the date of each year. The remaining columns indicate whether there is a frost depth (”1” indicates have a frost depth, ”0” indicates otherwise) for each year.

Month	Day	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
4	20	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0
4	21	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0
4	22	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0
4	23	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0
4	24	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
4	25	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
4	26	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	27	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	28	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
12	7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
12	8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
12	9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
12	10	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
12	11	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0
12	12	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
12	24	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
12	25	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
12	26	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
12	27	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
12	28	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
12	29	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
12	30	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Figure 3.4. Dates that have frost depth.The first two columns show the date of each year. The remaining columns indicate whether there is a frost depth (”1” indicates have a frost depth, ”0” indicates otherwise) for each year.

Note that in the merged data set (an example is shown in Figure. 3.2.), there is only soil temperature data of each depth for each date. However, to build a regression model for predicting frost depth, frost depth is needed as the response variable. Chapter 5 illustrates a linear interpolation method to estimate frost depth from the soil temperature values of each level.

We have two approaches to build a regression model for frost depth prediction. One approach is to build a model to predict soil temperature at each depth, and then uses linear interpolation method to calculate frost depth from the predicted soil temperature values at each depth. The other approach calculates soil frost depth values as the response variable first, and then build a model directly to predict frost depth. In Chapter 5 we will present techniques to build regression models for each approach, and analyze their pros and cons.

The original data set from year 1994 to year 2010 was split into a training data set, which includes data from year 1994 to year 2008, and a testing data set, which includes

data from year 2009 to year 2010. For both approaches, we built models on the training data set, and verified their effectiveness on the testing data set.

The training design matrix data as well as the testing design matrix data were normalized such that each variable's values are z-score [8] normalized, and the response variable of training and testing data sets are subtracted by their corresponding mean of each date in a year:

$$\begin{aligned} \tilde{X}_i &= (X_i - \bar{X})/S_x \\ \tilde{Y}_i &= Y_i - \bar{Y}_{\sigma_{date}(Y_i)} \end{aligned} \tag{1}$$

where $\tilde{X}_i$ and $\tilde{Y}_i$ are the i^{th} data point's feature vector and response respectively, $\bar{X}$ are the means of the design matrix, S_x is the estimation for standard deviation of the design matrix, $\sigma_{date}()$ is a date selection function (to find the date, which includes month and day, within a year. Jan. 1 for example, $\sigma_{date}(1/1)$ will select a subset of dates, including January 1. 1994, January 1. 1995, ... , January 1. 2010), and $\bar{X}_i$ and $\bar{Y}_{\sigma_{date}(Y_i)}$ are respectively the average feature vector and average response of a specific date selected by the date selection function $\sigma_{date}()$.

CHAPTER 4. LASSO REGRESSION ON PREDICTING FROST DEPTH

4.1. Introduction of Lasso Regression

Lasso regression was first introduced in 1994 by Tibshirani [5]. Lasso regression is first briefly summarized here in order to provide a context for our methods[1].

Suppose there is a population of p-dimensional vectors $\mathbf{X}$, where $\mathbf{X} \subset R^p$. Furthermore, there is a population of 1-dimensional real-valued responses $\mathbf{Y}$ ($\mathbf{Y} \subset$ R) corresponding to each $\mathbf{X}$. A general linear regression model to estimate the coefficients is as follows:

$$Y_i = X_i^{'}\beta + \varepsilon_i, \forall i = 1, ..., n \tag{2}$$

where X_i ($X_i = (1, x_1, x_2, ..., x_p)^{'}$) and Y_i are the i^{th} vector [2] and response respectively, β ($\beta = (\beta_0, \beta_1, ..., \beta_p)$) are coefficients, n is the number of vectors, and ε_i is the random error.

In matrix form:

$$\tilde{Y} = \mathbf{X}\beta + \tilde{\varepsilon} \tag{3}$$

where $\tilde{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ ... \\ Y_n \end{pmatrix}$ $\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ ... \\ X_n \end{pmatrix}$ $\tilde{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ ... \\ \varepsilon_n \end{pmatrix}$

and $f(\mathbf{X}) = \mathbf{X}\beta$ is denoted as the predicts for $\mathbf{Y}$.

In linear regression, β is usually estimated through minimizing the least squares objective function:

[1] The equations are mainly from [5]

[2] In the thesis, vector and data point are used interchangebly.

$$Z_n(\beta) = \frac{1}{n}(\tilde{Y} - \mathbf{X}\beta)^T(\tilde{Y} - \mathbf{X}\beta) \tag{4}$$

by first taking differentiation with respect to β:

$$\frac{\partial Z}{\partial \beta} = -2\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\beta) \tag{5}$$

and then by setting it to zero:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\tilde{Y} \tag{6}$$

The Lasso regression (Least Absolute Shrinkage and Selection Operator) [5] method applies a constraint on the sum of the first norm of coefficients when the least squares objective function is minimized:

$$Z_n(\beta) = \frac{1}{n}(\tilde{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta) + \lambda_n \sum_{j=1}^{p} |\beta_j| \tag{7}$$

which is usually called a penalized least square objective function. In Equation (7), λ_n is a tunning parameter.

Therefore, we can get the coefficients β with:

$$\hat{\beta} = \arg\min_{\beta} Z_n(\beta) \tag{8}$$

The idea behind the Lasso regression is basically to find a model complexity that optimally balances bias and variance.

The essence of Lasso regression lies in introducing some bias in the estimation for β so that the variance is reduced and hence the prediction error is decreased[1].

[1]Lasso regression does so through removing unimportant attributes. Thus it often used in model selection. For more details, please refer to [5].

In the Lasso regression objective function (Equation (7)), λ controls the amount of regularization. When $\lambda \to 0$ Lasso estimate is reduced to a linear regression model, and when $\lambda \to +\infty$ Lasso estimatel is reduced to a mean model (with interception only).

Note that Lasso regression objective function (Equation (7)) is a covex function, meaning that for each λ there will be only one set of coefficients β that minimizes the objective function (Equation (7)). There is no closed form solution to minimize the objective function. However, [5] provides a quadratic programming technique, and there is a R package (glmnet [9]) which is very convenient for solving the coefficients that minimize the objective function.

4.2. Lasso Regression for Correlated Errors

When there is a correlation between random errors ε_i in Equation (3), further study is needed to improve the results of Lasso regression. Note that to know whether a data set will have a correlation between random errors after modeling a Lasso regression, linear regression can be simply performed on the data set first, and then analyze the residues of the linear regression results.

Figure. 4.1. shows the linear regression residue plot for the training data set (data from year 1994 to year 2008). Figure. 4.2. shows a plot of auto correlation function (ACF) of the residues. Auto correlation describes the similarity between observations as a function of the time lag [10]. A clear pattern of the residues along the X-axis can be seen.

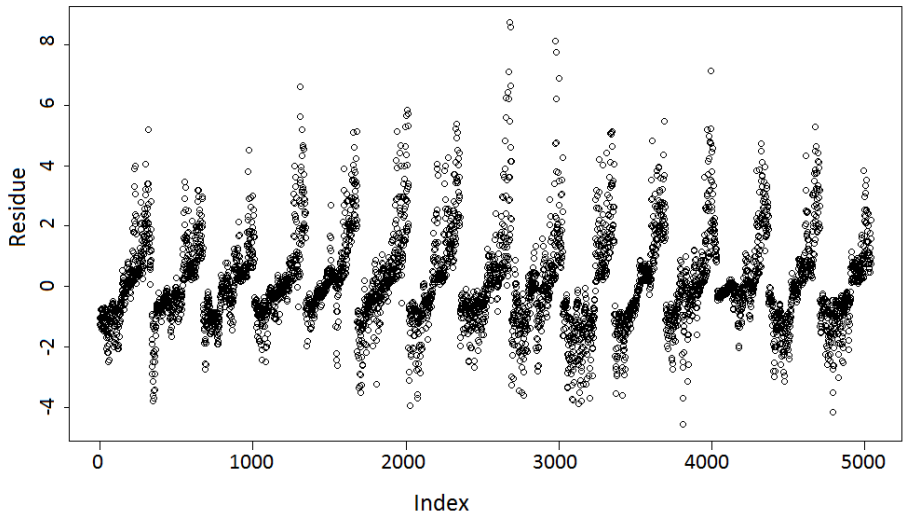

Figure 4.1. The linear regression residue plot on depth 10.

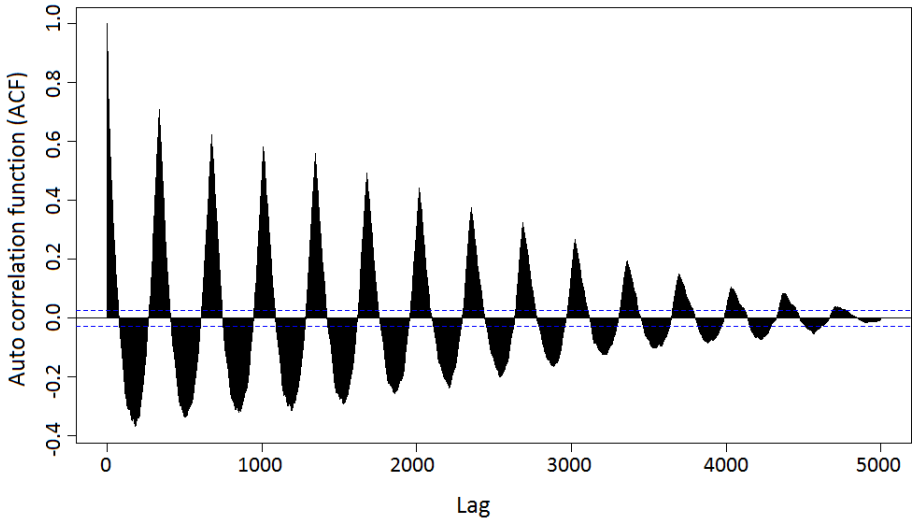

Figure 4.2. The plot of auto correlation function of linear regression residues on depth 10, which has a clear pattern of residues alone the X-axis.

The covariates for soil temperature in the data under consideration have high multicollinearity, which renders the classical ordinary least square estimates suffering from inflated standard error. The regular Lasso method can be applied to cure the multicollinearity. However, the random error in our model has temporal autocorrelation, thus the LASSO method for correlated data proposed by Gupta [6], "A not on the asymptotic distribution of Lasso estimator for correlated data", is used instead for estimation of all our soil temperature and frozen depth models.

Brief summary of applying Gupta's research [6] is as follows:

Assume $\sum = Cov(\varepsilon_1, ..., \varepsilon_n)$ is the covariance matrix of the random errors. From Equation (3), we can get:

$$(\sum)^{-1/2}\tilde{Y} = (\sum)^{-1/2}\mathbf{X}\beta + (\sum)^{-1/2}\tilde{\varepsilon} \tag{9}$$

Let $\tilde{Y}^* = (\sum)^{-1/2}\tilde{Y}$, $\mathbf{X}^* = (\sum)^{-1/2}\mathbf{X}$, and $\tilde{\varepsilon}^* = (\sum)^{-1/2}\tilde{\varepsilon}$, then we get:

$$\tilde{Y}^* = \mathbf{X}^*\beta + \tilde{\varepsilon}^* \tag{10}$$

Note that $Cov(\tilde{\varepsilon}^*) = I_n$, where I_n is an identity matrix.

Thus, the new regression coefficients β can be estimated as follows:

$$\hat{\beta} = \arg\min_{\beta} Z_n^{'}(\beta) \tag{11}$$

where :

$$\begin{aligned} Z_n^{'}(\beta) &= \tfrac{1}{n}(\tilde{Y}^* - \mathbf{X}^*\beta)^T(\tilde{Y}^* - \mathbf{X}^*\beta) + \lambda_n \sum_{j=1}^{p} |\beta_j| \\ &= \tfrac{1}{n}(\tilde{Y} - \mathbf{X}\beta)^T \sum{}^{-1}(\tilde{Y} - \mathbf{X}\beta) + \lambda_n \sum_{j=1}^{p} |\beta_j| \end{aligned} \tag{12}$$

where $\sum$ is the covariance matrix of the random errors. We use sample variance-autocovariance matrix of residues for $\sum$.

$$\hat{\sum} = \begin{bmatrix} \hat{\gamma}_0, \hat{\gamma}_1, & \cdots, & \hat{\gamma}_{(\mathrm{t}-1)} \\ \hat{\gamma}_1, \hat{\gamma}_2, & \cdots, & \hat{\gamma}_{(\mathrm{t}-2)} \\ \cdots & & \\ \hat{\gamma}_{(\mathrm{t}-1)}, \hat{\gamma}_{(\mathrm{t}-2)}, \cdots, \hat{\gamma}_{(0)} & & \end{bmatrix} \tag{13}$$

where:

$$\hat{\gamma}(h) = n^{-1} \sum_{k=1}^{n-h} (U_k - \bar{U}_n)(U_{k+h} - \bar{U}_n) \tag{14}$$

where $0 \leq h \leq n-1$ and $U = \mathbf{Y} - \mathbf{X}\hat{\beta}$. $\hat{\beta}$ is obtained iteratively by initializing the value of $\sum^{-1}$ as $\sum_0^{-1}$, which is the covariance matrix of residues of linear regression.

[6] also suggests taking the following value for λ_n:

$$\lambda_n = O(\frac{1}{\sqrt{n}\ln(n)}) \tag{15}$$

Figure. 4.3. summarizes that the entire algorithm is used to estimate β_s. The input data of the Algorithm 1 include design matrix of training data set and response variable. After normalizing the design matrix and response variable values, Algorithm 1 first calculates covariates β and correlation matrix $\sum$ using Linear regression. Then Algorithm 1 updates β and $\sum$ iteratively using Lasso regression until β converges or reaches maximum iteration.

Data: X; /* Design Matrix of training data set */
Data: Y; /* Response variable */
Result: β; /* Lasso regression model */
X = normalize(X); /* using Equation (1) */
Y = normalize(Y); /* using Equation (1) */
Calculate λ using Equation (15);
Calculate β and $\sum$ using Linear regression;
while β *not converge* && *iteration* $<$ *maxIteration* **do**
    Update β (via optim pakage) using Equations (11) and (12);
    Update $\sum$ using Equations (13) and (14);
    Increment iteration;
return β

Figure 4.3. Algorithm 1: Lasso regression.

CHAPTER 5. MODELING SOIL FROST DEPTH

As mentioned in Chapter 3, the first approach to predict frost depth is to model the soil temperature at each depth first, then use a simple linear interpolation method to calculate the frost depth. The following section describes this approach.

We have two further options to build models on predicting soil temperature at each depth. One is that we build different models on each depth separately. The other is to build a model on combined data set in which the depth is considered as a variable. One merit that the second option has over the first one is that the model of the second option can work on a new data set with new depth data, since the depth is regarded as an attribute. Thus, the first option would be prefered only in the case that the model built using the first option will have a better accuracy on predicting soil temperature, and hence better accuracy on frost depth later on[1].

Since the experiment setup for both options, including data preprocessing and programming, does not differs very much, both options will be tested separately.[2]

5.1. Modeling Soil Temperature First at Each Depth

Lasso Regression models were built using Algorithm 1 which is shown in Figure. 4.3. on predicting soil temperature for each depth. In this case, the response variable Y in Algoirthm 1 was the soil temperature at one depth. The design matrix X has the variables that are shown in Table 3.2. except $Year$, $Depth$, and $Temperature$. The source code is in Appendix. A problem using R in calculating the inverse matrix for the big matrix was found, so the Matlab (via R.matlab pakage) was used to do inverse matrix calculation.

[1] Actually through our experiments, we realized that the first option requires much less computer memory since it works on much smaller data size.

[2] Snow data is considered as one of the most important data in freeze and thaw cycle modeling[11]. In this research, we also tried to use snow data in "Modeling Soil Temperature First on All Depths" and "Modeling Soil Temperature First on Each Depth" methods. It turned out that the result did not improved obviously for the two methods with RMSE = 19.47 and 19.06 with and without snow data in "Modeling Soil Temperature First On Each Depth" and RMSE = 21.93 and 22.69 with and without snow data in "Modeling Soil Temperature First On All Depths". We do not know the reason, but statistically, we only want to include data that are significantly related to temperature changes. Therefore, the snow data were not used as one of the input variables in the models.

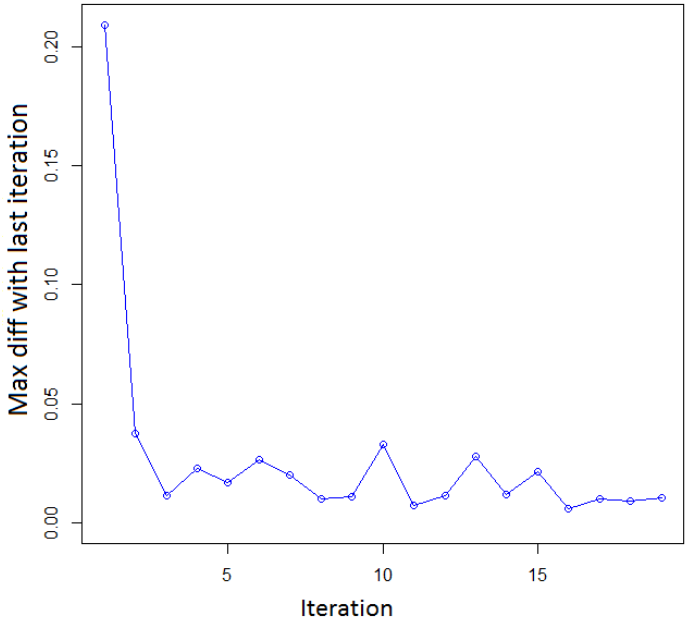

Figure 5.1. Convergence trend of beta on depth 10.

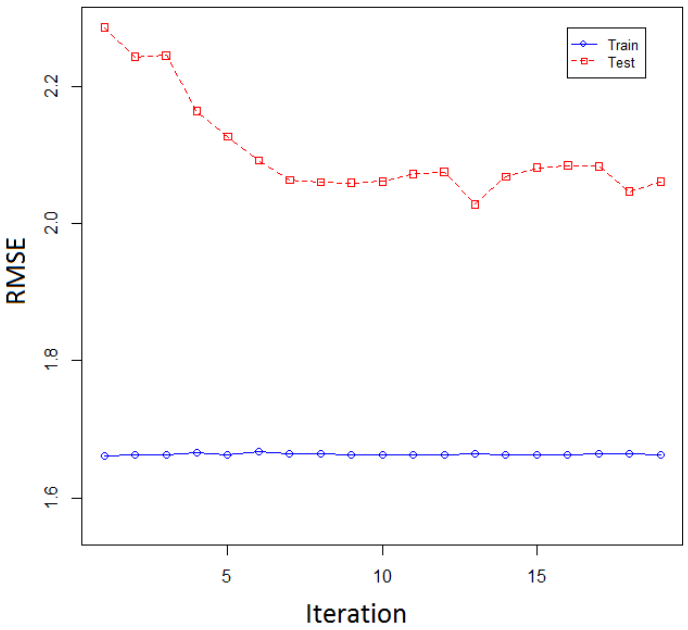

Figure 5.2. The root mean square errors (RMSE) of training data and testing data on depth 10.

Figure. 5.1. shows the convergence trend of β on depth 10 in Algorithm 1. The "maximum diff with last iteration" is defined as $max(\beta_i - \beta_{i-1})(i \in [2 \quad maxIteration])$. Figure. 5.2. shows the trend of root mean square errors (RMSE) of training data and testing data, respectively, using the βs that are calculated in each iteration in the $While$ loop of Algorithm 1 on depth 10. It can be seen that the RMSE of testing data generally improves over each iteration in Algorithm 1. The convergence trends of β and the trends of root mean square errors on other depths are quite similar to those in Figures. 5.1. and 5.2., thus not all of them are shown.

Tables 5.1., 5.2., and 5.3. summarizes models (β) for different depths. The values in parentheses is the standard errors. Notice that several covariates are equal to 0 in each model, highlighting the capability of Lasso regression to prune the models. The most significant covariates across all models are the air temperature ($AirTemp$), the total solar radiation ($Solar$), and the average dew point temperature (DP).

Table 5.1. Models for depths from 1cm to 30cm

	Depth1	Depth5	Depth10	Depth20	Depth30
Intercept	0.0850 (0.0009)	0 (0.0003)	0.0662 (0.0004)	0.0750 (0.0001)	0.0618 (0.0001)
ConvertedDay	0 (0.0006)	0 (0.0002)	0.0200 (0.0002)	0 (0.0002)	0 (0.0001)
Tmax	0.0823 (0.0017)	0.0450 (0.0013)	0.0214 (0.0005)	0 (0.0003)	0 (0.0007)
Tmin	0.0503 (0.0007)	0.0351 (0.0006)	0 (0.0008)	0 (0.0001)	0 (3.14E-05)
Tavg	0.0175 (0.0037)	0 (0.0051)	0 (0.0020)	0 (0.0012)	0.0167 (0.0010)
Tbs	-0.0635 (0.0008)	-0.0345 (0.0003)	-0.0111 (0.0004)	0 (0.0007)	0 (0.0002)
Tts	-0.0406 (0.0010)	-0.0286 (0.0009)	0.0297 (0.0009)	0 (0.0001)	0.0131 (7.70E-05)
WSavg	0 (0.0035)	0.0101 (0.0007)	0.0229 (0.0010)	0.03738 (0.0003)	0 (0.0006)
WSmax	0 (0.0018)	0.0137 (0.0006)	0 (0.0010)	0.0300 (0.0002)	0.0246 (0.0002)
WDavg	0 (1.87E-05)	0 (2.11E-05)	0 (2.44E-05)	0 (5.70E-06)	0 (1.31E-05)
Solar	-0.0520 (0.0011)	-0.0520 (0.0007)	-0.0508 (0.0009)	-0.0398 (0.0002)	-0.0359 (8.20E-05)
Rainfall	-0.0314 (0.0011)	-0.01842 (0.0011)	-0.0121 (0.0011)	0 (0.0003)	0 (0.0002)
DP	-0.0666 (0.0023)	-0.0537 (0.0019)	-0.0612 (0.0007)	-0.0635 (0.0007)	-0.0707 (0.0019)
WC	-0.0117 (0.0017)	0 (0.0012)	0 (0.0002)	0 (0.0003)	0 (0.0006)
Precip	-0.0358 (0.0005)	-0.0271 (0.0004)	-0.0196 (0.0007)	-0.0112 (4.37E-05)	0 (9.47E-05)
AirTemp	0.0881 (0.0023)	0.0867 (0.0069)	0.0885 (0.0020)	0.08112 (0.0014)	0.0690 (0.0003)

Table 5.2. Models for depths from 40cm to 125cm

	Depth40	Depth50	Depth80	Depth100	Depth125
Intercept	0.0450 (0.0002)	0.03335 (0.0001)	0 (0.0001)	0 (3.11E-05)	0 (2.16E-05)
ConvertedDay	0 (9.46E-05)	0 (0.0001)	0 (0.0004)	0 (7.22E-05)	0 (6.24E-05)
Tmax	0 (0.0002)	0 (0.0007)	0 (3.50E-05)	0 (0.0002)	0 (9.07E-05)
Tmin	0 (0.0001)	0 (0.0002)	0 (0.0001)	0 (7.96E-05)	0 (4.23E-05)
Tavg	0 (0.0008)	0 (0.0010)	-0.0342 (0.0005)	0 (0.0006)	0 (0.0005)
Tbs	0 (0.0002)	0 (0.0006)	0 (4.73E-05)	0 (6.41E-05)	0 (9.12E-05)
Tts	0.01690 (0.0002)	0.0147 (0.0002)	0.0115 (4.37E-05)	0 (7.20E-05)	0 (2.70E-05)
WSavg	0 (0.0004)	0 (1.76E-05)	0 (6.62E-05)	0.0124 (0.0003)	0 (0.0002)
WSmax	0.01790 (0.0002)	0.01579 (0.0006)	0.0148 (0.0007)	0 (0.0002)	0 (0.0001)
WDavg	0 (3.47E-06)	0 (1.04E-05)	0 (2.21E-05)	0 (1.04E-05)	0 (4.07E-06)
Solar	-0.0244 (0.0002)	-0.0209 (0.0001)	-0.0152 (0.0001)	-0.0129 (8.33E-05)	-0.0133 (1.23E-05)
Rainfall	0 (0.0009)	0 (0.0004)	0 (0.0004)	0 (0.0003)	0 (0.0001)
DP	-0.0594 (0.0004)	-0.0490 (0.0009)	-0.0698 (0.0002)	-0.0147 (0.0001)	-0.0139 (0.0001)
WC	0 (0.0002)	0 (0.0003)	0 (0.0001)	0 (0.0001)	0 (0.0001)
Precip	0 (0.0002)	0 (0.0001)	0 (3.31E-05)	0 (5.38E-05)	0 (4.99E-05)
AirTemp	0.0695 (0.0008)	0.0533 (3.93E-05)	0.1053 (0.0005)	0.0210 (0.0005)	0.0196 (0.0003)

Table 5.3. Models for depths from 175cm to 250cm

	Depth175	**Depth200**	**Depth250**
Intercept	0 (4.17E-05)	0 (2.22E-05)	0 (1.03E-06)
ConvertedDay	0 (7.55E-05)	0 (4.79E-05)	0 (3.17E-05)
Tmax	0 (6.22E-05)	0 (0.0002)	0 (0.0002)
Tmin	0 (6.31E-05)	0 (3.51E-05)	0 (4.08E-05)
Tavg	0 (0.0004)	0 (0.0007)	0 (0.0008)
Tbs	0 (5.81E-05)	0 (4.43E-05)	0 (0.0003)
Tts	0 (7.07E-05)	0 (4.83E-05)	0 (4.92E-05)
WSavg	0 (8.14E-05)	0 (0.0001)	0 (0.0005)
WSmax	0 (0.0002)	0 (0.0001)	0 (0.0001)
WDavg	0 (3.43E-06)	0 (1.03E-06)	0 (1.44E-06)
Solar	-0.0136 (4.59E-05)	-0.0122 (7.52E-05)	-0.0114 (0.0002)
Rainfall	0 (2.31E-05)	0 (2.74E-05)	0 (0.0001)
DP	-0.0128 (7.41E-05)	-0.0115 (9.23E-05)	0 (0.0006)
WC	0 (9.37E-05)	0 (9.98E-05)	0 (0.0003)
Precip	0 (6.02E-05)	0 (2.23E-05)	0 (2.43E-05)
AirTemp	0.01857 (0.0003)	0.01677 (0.0007)	0.0121 (0.0004)

Figures 5.3. and 5.4. show the residue plot and ACF plot for residue , respectively, after Lasso regression on depth 10. From these two figures, It can be seen that the correlation among residues is greatly reduced in comparison with that of the linear regression shown in Figures 4.1. and 4.2..

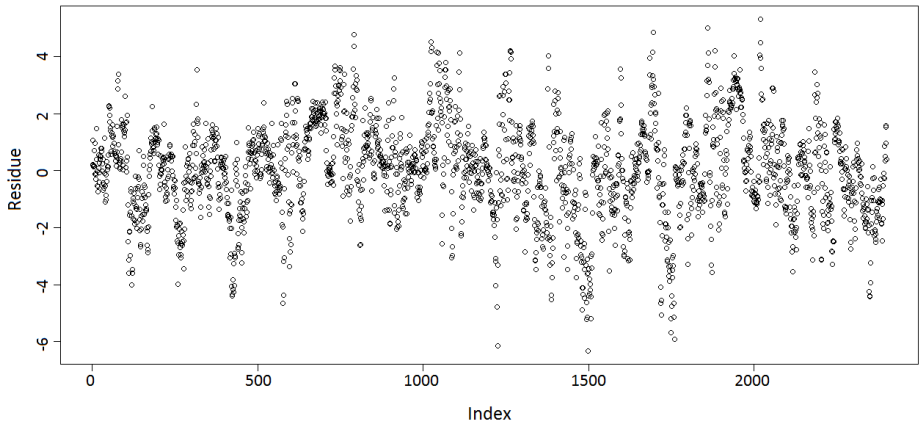

Figure 5.3. Plot of residue after Lasso regression on depth 10.

After models were built to predict soil temperature on each depth, the frost depth could hence be calculated straightforwardly. As mentioned before, the frost depth is defined as the depth at which the soil temperature is 0°C, and above which the soil temperature is greater than 0°C. Take April 15. 2002 for example, as shown in Figure 5.5.. The frost depth is slightly larger than 40 cm.

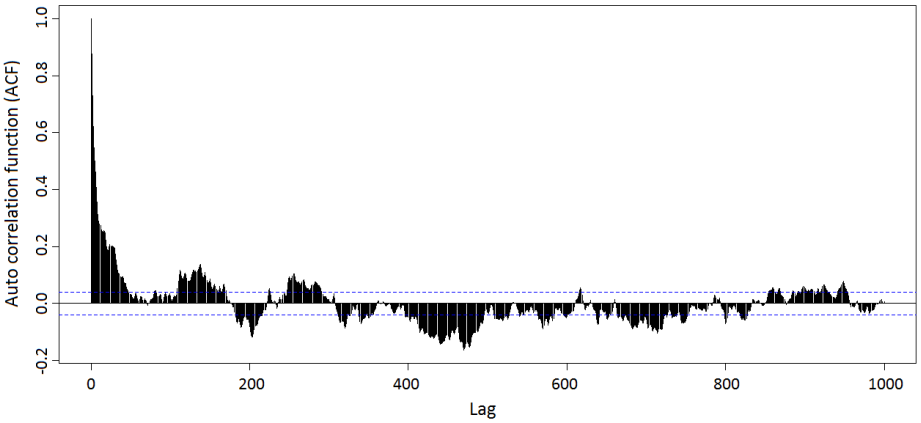

Figure 5.4. ACF plot for residue after Lasso regression on depth 10.

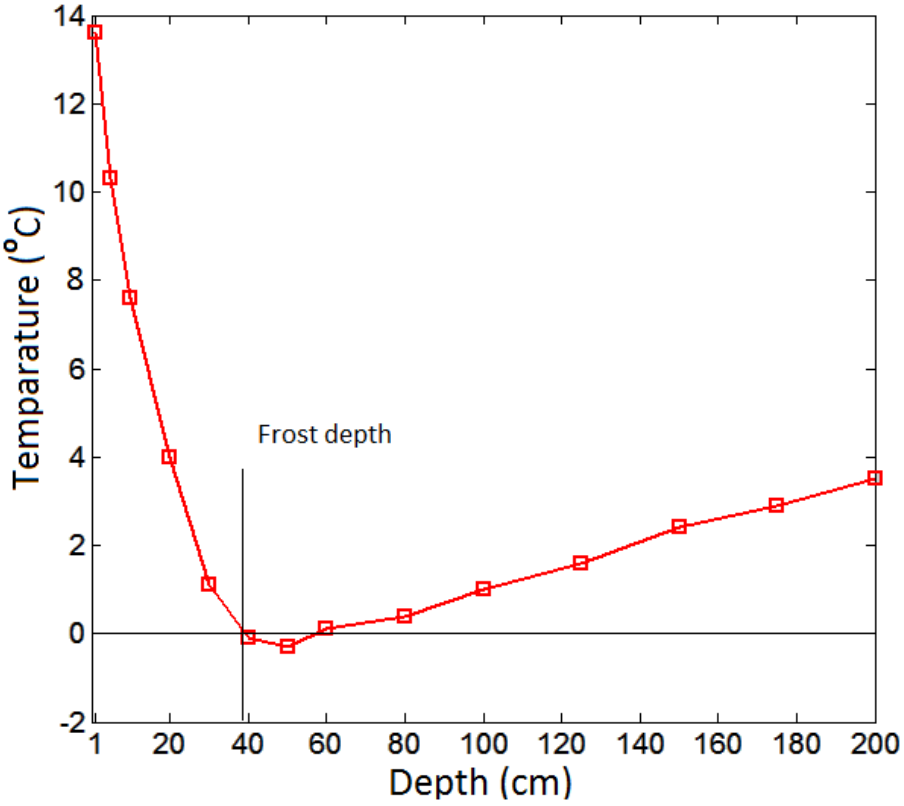

Figure 5.5. Frost depth on April 15. 2002.

Equation 16 shows a linear interpolation method to calculate frost depth:

$$\begin{aligned} & D_f = \max(D_i + \tfrac{Y_i}{Y_i - Y_{i+1}} \times (D_{i+1} - D_i), 0) \\ & s.t. \qquad \forall \mathrm{j} \leq \mathrm{i}, \quad Y_j > 0, \quad \mathrm{Y}_{i+1} \leq 0 \end{aligned} \tag{16}$$

where D_f is the frost depth, Y_i and Y_{i+1} are the temperatures of the i^{th} and $(i+1)^{th}$ depths, respectively. D_i and D_{i+1} are the values of the i^{th} and $(i+1)^{th}$ depths, respectively.

It is possible that there may exist a frost depth, while all the depth temperatures are greater than 0^oC. The linear interpolation method will not work in such cases.

5.2. Modeling Soil Temperature First on All Depths

To build Lasso Regression models for predicting soil temperature on all depths, the response variable Y in Algorithm 1 is still the soil temperature. The design matrix X has the variables that are shown in Table 3.2. except $Year$ and $Temperature$. The source code

for this case is only slightly different from the one in Appendix, so we omit the code in the thesis.

Figure. 5.6. shows the convergence trend of β in Algorithm 1. Figure. 5.7. shows the trend of root mean square errors (RMSE) of training data and testing data respectively using the βs that are calculated in each iteration in the $While$ loop of Algorithm 1. It can be seen that the RMSE of testing data slightly improves over each iteration in Algorithm 1.

Table 5.4. summarizes the model (β) on all depth.

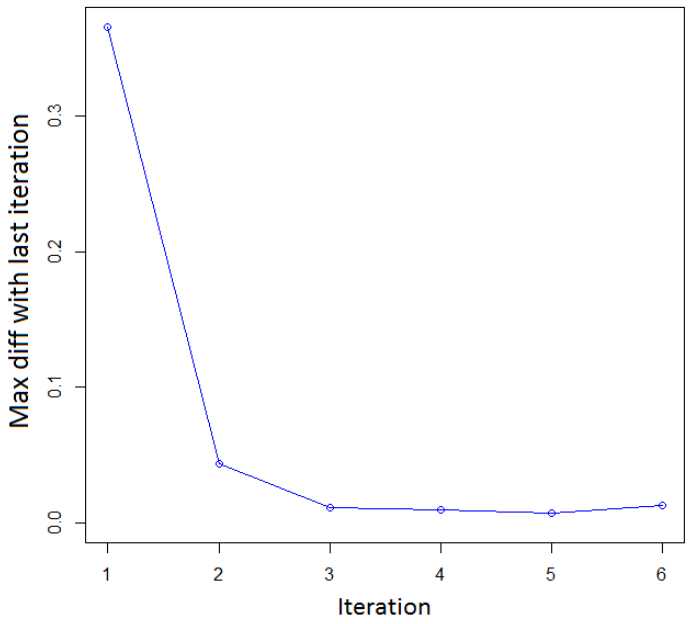

Figure 5.6. Convergence trend of beta on all depth.

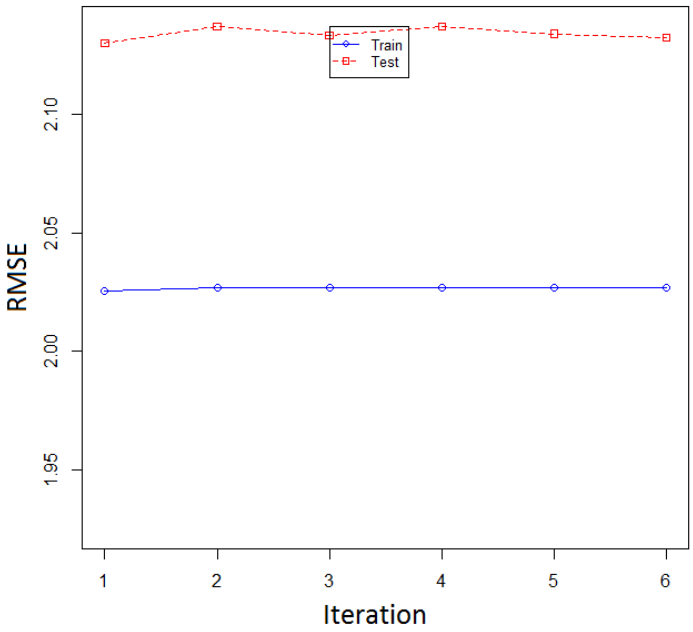

Figure 5.7. The root mean square errors (RMSE) of training data and testing data on all depth.

Table 5.4. The model on all depths

	Depth Combined
Intercept	0.2079 (0.0010)
ConvertedDay	0 (0.0002)
Tmax	0.0164 (0.0003)
Tmin	0 (2.11E-05)
Tavg	0.0154 (0.0001)
Tbs	0 (0.0006)
Tts	0.0145 (0.0005)
WSavg	0.0518 (0.0018)
WSmax	0.0388 (0.0005)
WDavg	0 (1.88E-05)
Solar	-0.0127 (0.0004)
Rainfall	0 (0.0008)
DP	0.0111 (0.0007)
WC	0.0126 (0.0002)
Precip	0 (0.0008)
Depth	0.0107 (0.0002)
AirTemp	-0.0108 (0.0004)

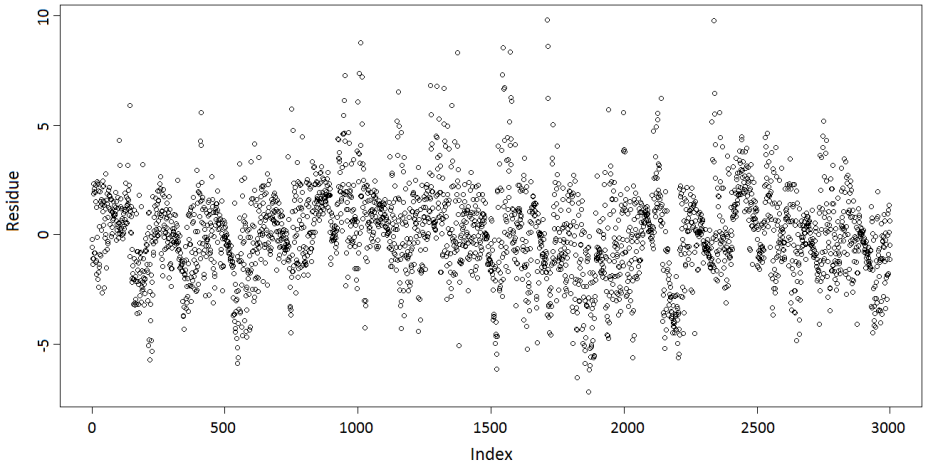

Figure 5.8. Plot of residue after Lasso regression on the whole dataset.

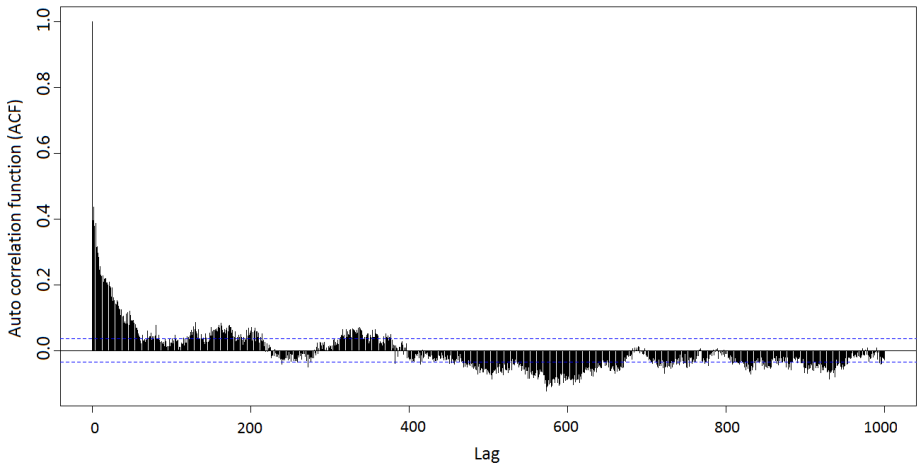

Figure 5.9. ACF plot for residue after Lasso regression the whole dataset.

Figures 5.8. and 5.9. show the residue plot and ACF plot for residue respectively after Lasso regression on the whole dataset. From these two figures, we can see that the correlation among residues is also greatly reduced in comparison with that of linear regression, as in Figures 4.1. and 4.2..

5.3. Modeling Directly on Frost Depth

To build models directly on frost depth, the frost depth is calculated on the training data first using linear interpolation method, and the frost depth is considered as the response variable. However, as it was mentioned before, only a fraction of days within a year had frost depth, as shown in Figures 3.2., 3.3., and 3.4.. Besides, the dates having frost depth of each year are also not consistent. Therefore, the subset of dates that most years have a frost depth was selected. In Figure. 3.3., this subset of dates are underlined.

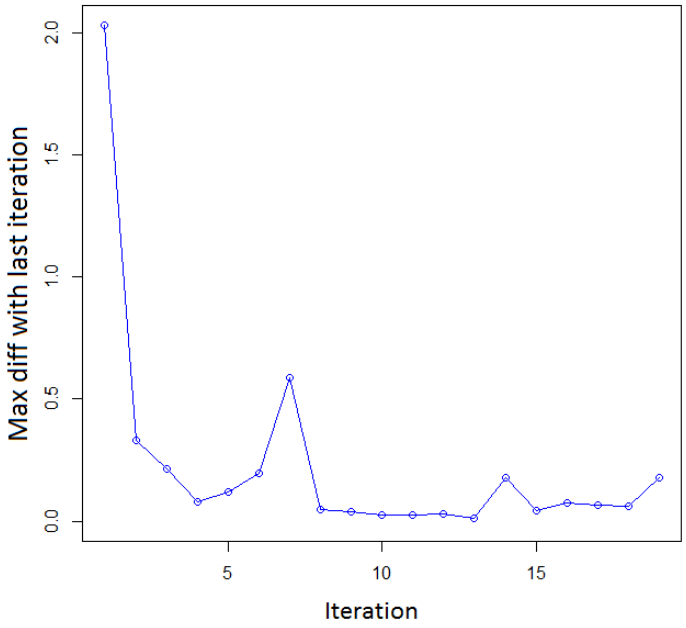

Figure 5.10. Convergence trend of beta when modeling frost depth directly.

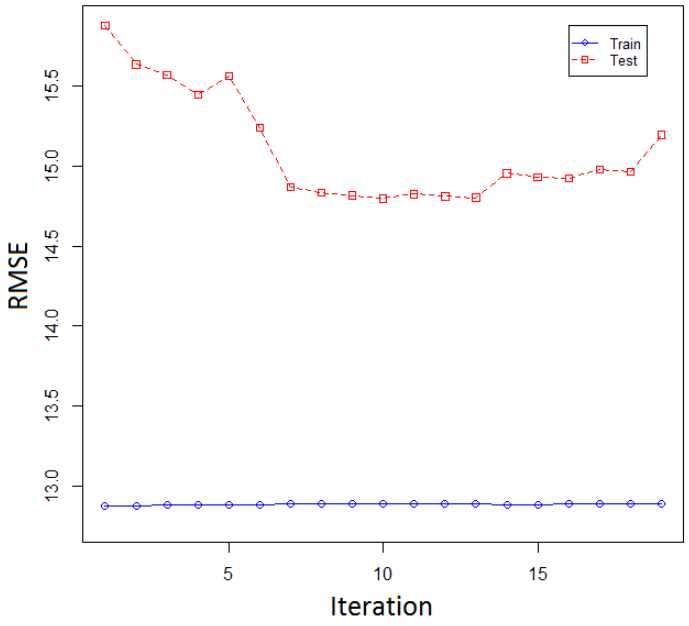

Figure 5.11. The root mean square errors (RMSE) of training data and testing data when modeling frost depth directly.

Figure. 5.10. shows the convergence trend of β when modeling frost depth directly. Figure. 5.11. shows the trend of root mean square errors (RMSE) of training data and testing data using the βs that are calculated in each iteration in the $While$ loop of Algorithm 1. We can see that the RMSE of testing data generally improves over each iteration in Algorithm 1.

Table 5.5. summarizes the model on frost depth directly.

Figure. 5.12. shows a screenshot of the application for data preprocessing and models training. Figure. 5.13. shows a screenshot of the application for model comparison.

Table 5.5. The model on frost depth directly

	Depth Combined
Intercept	-0.8184 (0.0517)
ConvertedDay	0 (0.0940)
Tmax	1.3562 (0.0683)
Tmin	0.4949 (0.1984)
Tavg	-2.8945 (0.0777)
Tbs	-0.7594 (0.5139)
Tts	-1.0636 (0.0523)
WSavg	-1.0840 (0.0193)
WSmax	0.4459 (0.0054)
WDavg	-0.0293 (0.0843)
Solar	-0.4337 (0.0893)
Rainfall	-1.4478 (0.2411)
DP	-0.0243 (0.0893)
WC	0.8386 (0.1079)
Precip	1.1080 (1.2724)
AirTemp	1.1062 (0.2630)

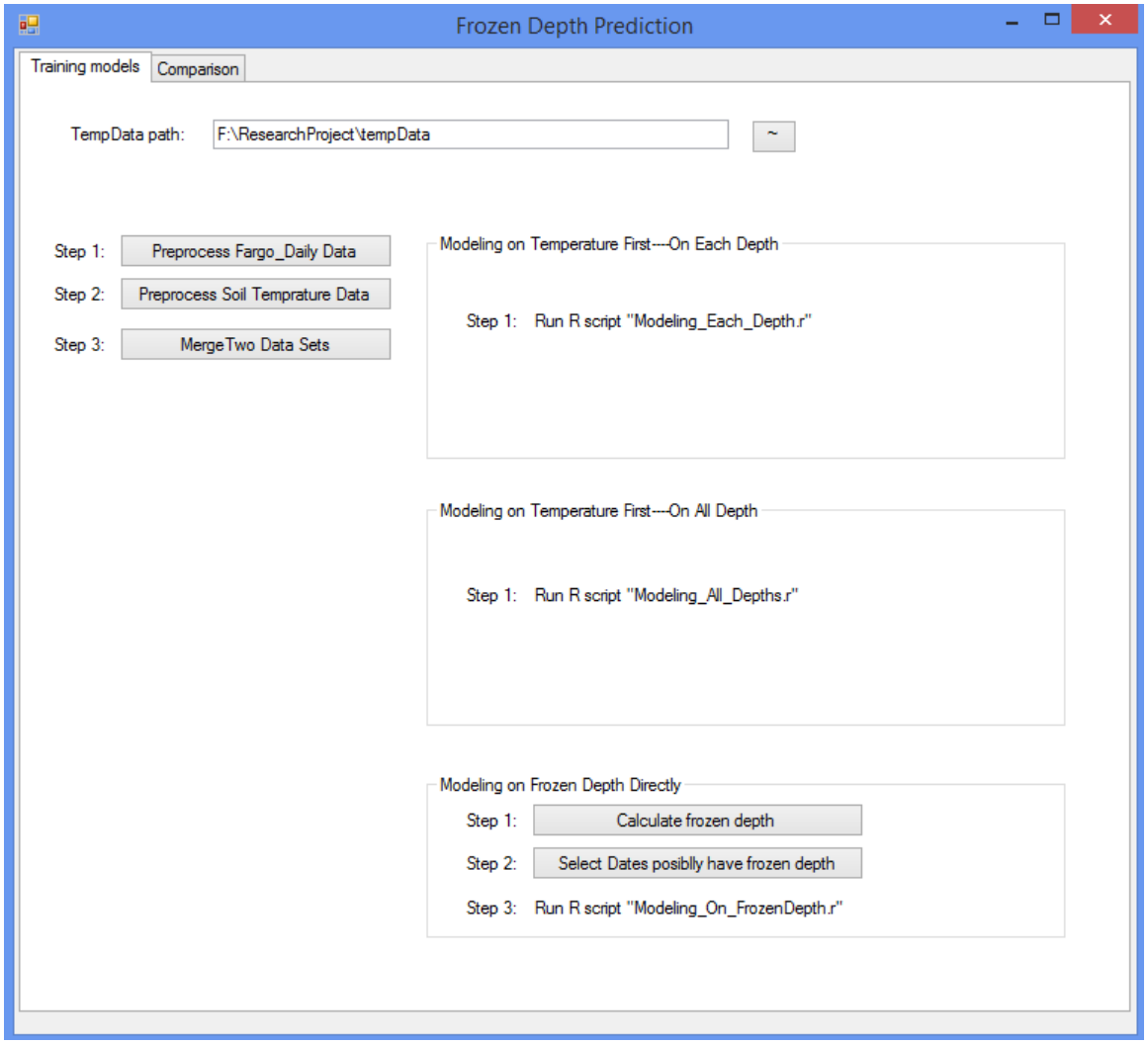

Figure 5.12. Screenshot of the application for data preprocessing and model training.

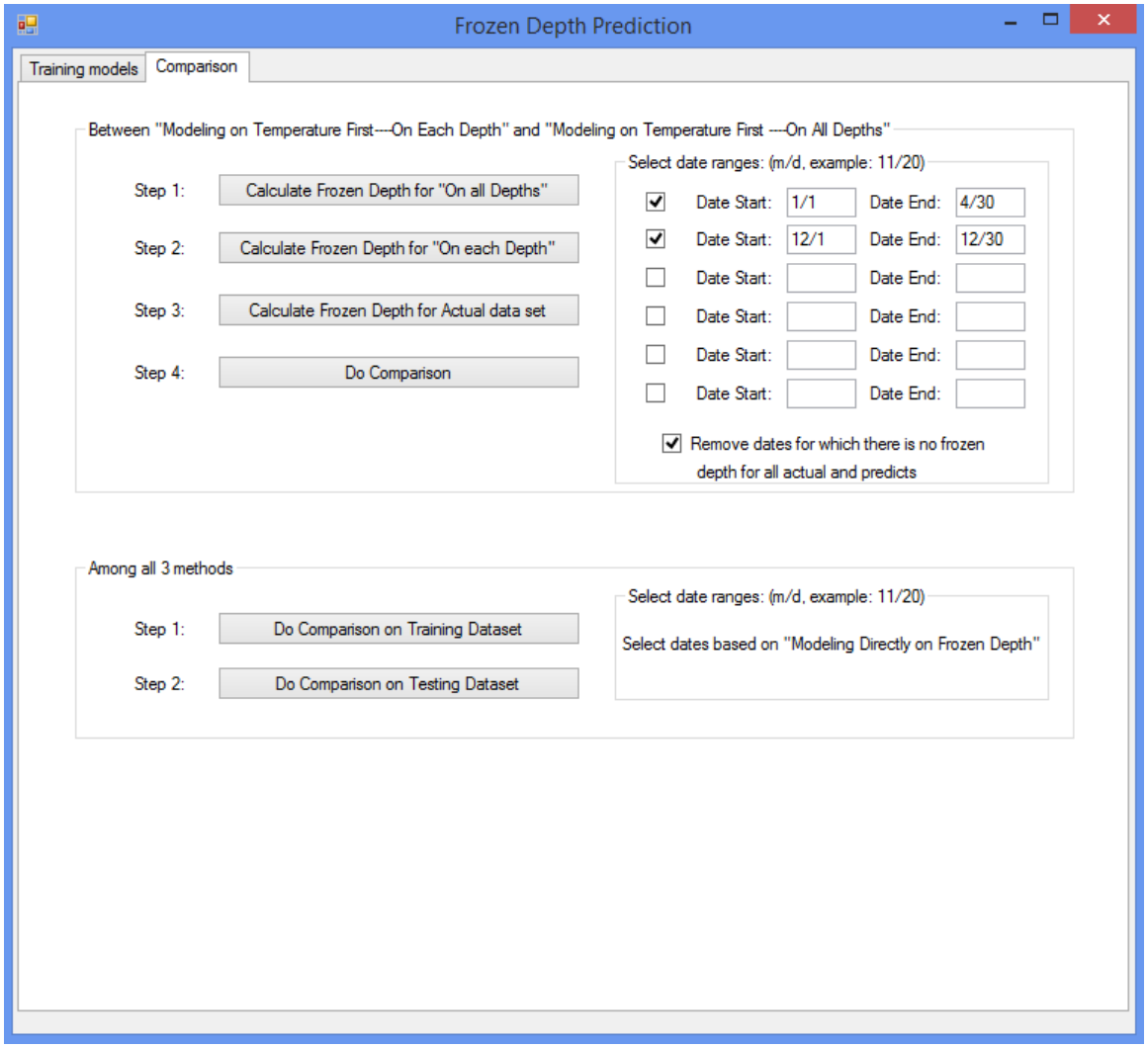

Figure 5.13. Screenshot of the application for model comparison.

CHAPTER 6. COMPARISON

6.1. Comparison Between Two Options of "Modeling Soil Temperature First" Methods

In Sections 5.1 and 5.2, two options were presented to predict frost depths by first modeling soil temperatures. These two options can work on any dates for which there is even no frost depths. The first option, "Modeling Soil Temperature First at Each Depth", creates one regression model on each depth. While the second option, "Modeling Soil Temperature First on All Depths", however, creates only one model for all depths.

As we already discussed, there is only a very small fraction of dates that have a frost depth. Thus, it would make little sense to select all dates within a year to compare the performance of these two options. For example, we would expect that the frost depths do not exist[1] using models built with these two options on summer days. Including such dates does not help to differentiate the performance of both models. Therefore, we select the subset of dates using the following equation:

$$\cup(\sigma_{date}(Y > 0), \sigma_{date}(\tilde{Y_1} > 0), \sigma_{date}(\tilde{Y_2} > 0)) \tag{16}$$

where Y is the actual frost depths, $\tilde{Y_1}$ is the predicted frost depths of the first option models, $\tilde{Y_2}$ is the predicted frost depths of the second option model, $\cup$ is the union function, and $\sigma_{date}()$ is a date selection function. For example, $\sigma_{date}(Y > 0)$ selects the subset of dates for which the actual predictions are greater than 0.

In this study, root mean square errors (RMSE) [12] is used to compare different methods. Table 6.1. shows the comparison of root mean square errors (RMSE) between the two options of "Modeling Soil Temperature First" methods.

From Table 6.1., we can see that "Modeling Soil Temperature First on Each Depth" outperforms "Modeling Soil Temperature First on All Depth". Besides, as mentioned

[1] We set frost depth to 9999 if it does not exist.

Table 6.1. Comparison of RMSE between two options of "Modeling Soil Temperature First" methods

	RMSE
Modeling Soil Temperature First on Each Depth	19.06
Modeling Soil Temperature First on All Depths	22.69

before, "Modeling Soil Temperature First on Each Depth" needs much less computer memory when building training models since it works on much less data sets. However, models that are built with the option "Modeling Soil Temperature First on All Depth" are more robust in future data.

6.2. Comparison Among All Methods

To compare the performance of all three methods, namely "Modeling Soil Temperature First on All Depths", "Modeling Soil Temperature First on Each Depth ", and "Modeling Soil Temperature Directly on Frost Depth", we select the subset of dates that is used by "Modeling Soil Temperature Directly on Frost Depth" since it is much smaller than the other two.

Table 6.2. shows the comparison of the root mean square errors between the two options of "Modeling Soil Temperature First" methods.As we can see from Table 6.2., "Modeling Soil Temperature Directly on Frost Depth" significantly outperforms the other two methods.

Table 6.2. Comparison of RMSE among all three methods

	rmse
Modeling Soil Temperature First on Each Depth	18.82
Modeling Soil Temperature First on All Depths	20.39
Modeling Soil Temperature Directly on Frost Depth	10.64

Figure. 6.1. shows profiles of the actual frost depths and the predicted frost depths by the "Modeling Soil Temperature Directly on Frost Depth" method.

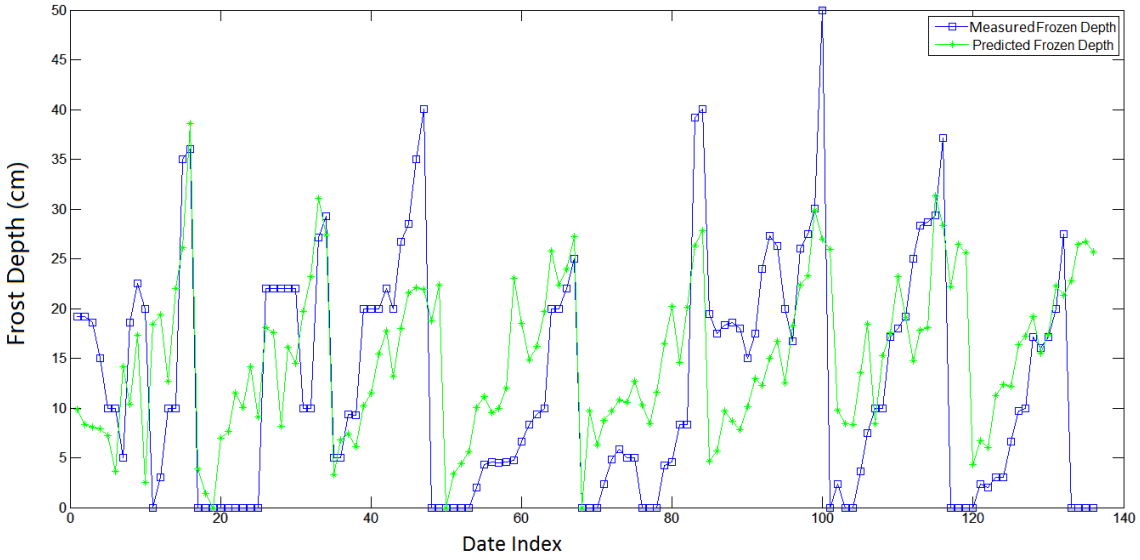

Figure 6.1. Measured and predicted frost depths.

6.3. Testing on New Data

In order to verify the three methods' capabilities of predicting frost depths on future data, ten days in 2012 were selected to test the three methods. In the ten days, there are six days (in March 2012) which have frost depths, and the other four days which have no frost depth were randomly selected in April 2012 . Table 6.3. shows the selected subset of dates.

Table 6.3. A subset of 10 random days in 2012

March 11	March 12	March 13	March 14	March 15
March 16	April 4	April 7	April 15	April 22

The comparison results are shown in Table 6.4.. As we can see from Table 6.4., "Modeling Soil Temperature Directly on Frost Depth" still outperforms the other two methods, and "Modeling Soil Temperature First on Each Depth" methods is slightly better than "Modeling Soil Temperature First on All Depths" method.

Table 6.4. Comparison of RMSE among all three methods

	rmse
Modeling Soil Temperature First on Each Depth	21.0
Modeling Soil Temperature First on All Depths	25.46
Modeling Soil Temperature Directly on Frost Depth	20.89

Table 6.5. shows the measured frost depth as well as the predicted frost depths of the three methods[1].

Table 6.5. Measured and predicted frost depths

date	**measured frost depth**	**Method 1**	**Method 2**	**Method 3**
March 11	8.89	0.0	0.0	1.05
March 12	21.67	0.0	0.0	1.18
March 13	26.67	4.024	56.27	13.16
March 14	34.0	1.45	0.0	5.17
March 15	37.14	4.54	0.0	3.33
March 16	45	16.97	1.73	9.10
April 4	0.0	0.0	0.0	0.0
April 7	0.0	0.0	0.0	0.0
April 15	0.0	0.0	0.0	0.0
April 22	0.0	0.0	0.0	0.0

Method 1:Modeling Soil Temperature First on Each Depth
Method 2:Modeling Soil Temperature First on All Depths
Method 3:Modeling Soil Temperature Directly on Frost Depth

Notice that all three methods successfully predicted non-frost depths in the randomly selected four days.

[1]If frost depth does not exist, a value of 0 is used to avoid unreasonable RMSE values

```
Data: M1;        /* The model from option "Modeling Soil
     Temperature First on each Depth" */
Data: M2;        /* The model from option "Modeling Soil
     Temperature First on All Depths" */
Data: M3; /* The model from option "Modeling Directly on
     Frost Depth" */
Data: x;                                  /* A new data point */
Result: F;                              /* Predicted Frost Depth */
if x has new depth temperature values then
    F = Apply M2 on x.
else
    if x's date is among the subset of dates used in M3 then
        F = Apply M3 on x.
    else
        F = Apply M1 on x.
return F
```

Figure 6.2. Algorithm 2: Combining three methods.

6.4. Combining Three Methods

As we have discussed above, the three frost-depth-predicting methods each have their advantages and disadvantages. Thus we further developed an algorithm, as shown in Algorithm 2 in Figure. 6.2., to combine the three methods, so that the combined method is more robust. In Algorithm 2, $M1$ is the model trained from option "Modeling Soil Temperature First on each Depth", $M2$ is the model trained from option "Modeling Soil Temperature First on All Depths", and $M3$ is the model trained from "Modeling Directly on Frost Depth". Given a new data point x, Algorithm 2 first determine if there is new depth temperature values or not. If there are new temperature values, then it applies $M2$ on the new data point x to calculate the frost depth F. Otherwise Algorithm 2 further determines if the new data point's date is among the subset of dates that are used in $M3$. If so it applies $M3$ on new data point x to get predicted frost depth F. Otherwise it applies $M1$ on x to estimate frost depth F.

By combining the three frost-depth-predicting methods, we not only retain the high prediction acuracy from option "Modeling Directly on Frost Depth" and "Modeling Soil Temperature First on Each Depth", but also gain the robustness of option "Modeling Soil Temperature First on All Depths".

CHAPTER 7. CONCLUSION

In this project, frost depth was modeled using weather and soil temperature data. Lasso regression technique was mainly used in modeling frost depth. Through analysis, the correlation was identified among residues after applying regular Lasso regression to the data. Guptas' research [6] "A note on the asymptotic distribution of Lasso estimator for correlated data", has been used in improving the Lasso regression in this project.

Using Lasso regression with residue correlation, we developed three methods to model frost depth, namely "Modeling Soil Temperature First on All Depths", "Modeling Soil Temperature First on Each Depth", and "Modeling Directly on Frost Depth". Amongthe three methods, "Modeling Directly on Frost Depth" achieves the highest accuracy using root mean square error measurement, while "Modeling Soil Temperature First on Each Depth" consumes the least computer memory during modeling training phase and "Modeling Soil Temperature First on All Depth" is the most robust method considering future data having different depth temperatures.

Finally, we also presented an algorithm to combine the three methods such that we not only retain the high accuracy of the "Modeling Directly on Frost Depth" and the "Modeling Soil Temerature First on Each Depth" methods but also gain the robustness of "Modeling Soil Temperature First on All Depths" method.

REFERENCES

[1] S. P. Farrington, "Frost penetration prediction using simulation with gis," in *The 22nd Annual Esri International User Conference*, 2002.

[2] S. Thordarson, N. Jonasson, E. Sveinbjomsson, A.H.Thorolfsson, and G.O.Bjomsson, "Real-time frost depth forecast model for thaw-induced axle load limitation management," in *Proceeding of XIIIth PIARC Winter Road Congress*, (Quebec), Feb 2010.

[3] H. Soliman, S. Kass, and N. Fleury, "A simplified model to predict frost penetration for manitoba soils," in *Annual Conference of the Transportation Association of Canada*, (Toronto), 2008.

[4] J. Lee, H. S. Kim, and Y. S. Kim, "Estimation of frost depth in south korea," *American Society of Civil Engineers*, 2013.

[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[6] S. Gupta, "A note on the asymptotic distribution of lasso estimator for correlated data," *The Indian Journal of Statistics*, vol. 74, pp. 10–28, 2012.

[7] "Fargo station deep soil temperatures, http://www.ndsu.edu/ndsco/soil/farg/farg.htm."

[8] E. Kreyszig, *Applied Mathematics*. Wiley Press, 1979.

[9] "glmnet pakage. http://cran.r-project.org/web/packages/glmnet/index.html."

[10] "Auto correlation function, http://en.wikipedia.org/wiki/autocorrelation."

[11] A. J. Phillips and N. K. Newlands, "Spatial and temporal variablility of soil freeze-thaw cycling across southern alberta, canada," *Agricultural Sciences*, vol. 2, pp. 392–405, 2011.

[12] "Root mean square error, http://en.wikipedia.org/wiki/root-mean-square_deviation."

APPENDIX. SOURCE CODE: PREDICTING SOIL TEMPERATURE

```
##################################################
##############1. Preprocessing#################
##################################################
session <- "1"
Date_Path <- "F : /ResearchProject/tempData/preprocessed"
####load libraries
OK=require(glmnet)
if(!OK) {
    install.packages(repos="http://cran.r-project.org", "glmnet")
}
library(glmnet)
OK=require(R.matlab)
if(!OK) {
    install.packages(repos="http://cran.r-project.org", "R.matlab")
    install.packages(repos="http://cran.r-project.org", "R.oo")
}
library(R.matlab)
library(R.oo)
setwd(Date_Path)
X <- read.csv("merged3.csv", strip.white = TRUE)
#######
##1.1##preprocess data so that each year would have a same subset of dates
# This probably not a good approach. Filling missing values with average would be better.
#######
#1993 and 2011 data are not complete. So discard them first
X <- X[X$Year! =1993,]
X <- X[X$Year! =2011,]
years <- unique(X$Year)
converted_value <- X$Month *12*31 + X$Day #*12 is to make sure no two different MonthDay pair generate
the same value.
common_subset <- converted_value[(X$Year==years[1])]
for(yearIndex in 2:length(years))
{
  common_subset <- intersect(common_subset,converted_value[X$Year==years[yearIndex]])
}
```

```
index <- rep(FALSE,dim(X)[1])
for(yearIndex in 1:length(years))
{
  t <- grep(years[yearIndex],X$Year)
  i=1
  j=1
  while(i <= length(t)  j <= length(common_subset))
  {
    if(converted_value[t[i]]<common_subset[j])
    {
      i <- i+1
    }
    else if(converted_value[t[i]] == common_subset[j])
    {
      index[t[i]] <- TRUE
      i <- i+1
    }
    else
    {
      j <- j+1
    }
  }
}
X <- X[index,]
rm(t)
rm(index)
rm(common_subset)
#######
##1.2##Select levels to test
#######
#depths_selected = c(1,5,10,20,30,40,50,60,80,100,125,150,175,200,250)
#depth60 and depth80 have 1920 data points, depth150 has 2080 data points
#all other depth have 2400 data points.
depths_selected = c(1,5,20,30)
depths_selected = depths_selected[as.numeric(session)]
index <- rep(FALSE, dim(X)[1])
for(depth in depths_selected)
{
```

```
  index <- index| (X$Depth==depth)
}
X <- X[index,];
rm(index)
#######
##1.3##Select days to test
#######
days_selected = seq(1,31,by=1)
index <- rep(FALSE, dim(X)[1])
for(day in days_selected)
{
  index <- index | (X$Day==day)
}
X <- X[index,];
rm(index)
#######
##1.4##Seperate training/testing data sets
#######
index <- rep(FALSE, dim(X)[1])
train_years <- seq(1994,2008,by=1)
test_years <- c(2009,2010)
for(year in test_years)
{
  index <- index | (X$Year==year)
}
Test <- X[index,];
Train <- X[!index,];
rm(X)
#######
##1.5##Withdraw Temperature
#######
Train_Y <- Train$Temperature
Train$Temperature <- NULL
Test_Y <- Test$Temperature
Test$Temperature <- NULL
dt<- Train$Month*12*31+Train$Day
dt_test<- Test$Month*12*31+Test$Day
#######
```

```
##1.6##remove mean w.r.t. each day of a month.
#######
#The depth in train and test will be normalized. keep a copy of depth for later indexing
Train_Depth <- Train$Depth
Test_Depth <- Test$Depth
if (length(depths_selected) == 1)
{
  Train$Depth<-NULL
  Test$Depth<-NULL
}
tt <- seq(0, dim(Train)[2]*length(days_selected)*12)
Means <- seq(1, dim(Train)[2])
Stds <- seq(1, dim(Train)[2])
MeansResponse <- array(tt, dim=c(1,length(days_selected),12))
Means[3:dim(Train)[2]] <- colMeans(Train[, 3:dim(Train)[2]])
for (i in 3:dim(Train)[2])
{
  ttt <- sd(unlist(Train[, i]))
  Stds[i] <- ttt
}
Stds[1] <- 1
Stds[2] <- 1
for (month in 1:12)
{
  for (dayIndex in 1:length(days_selected))
    {
      tt <- (Train$Month == month  Train$Day == days_selected[dayIndex])
      if(length(grep(TRUE,tt))==0)
      {
        next
      }
      MeansResponse[1,dayIndex,month] <- mean(Train_Y[tt])
      Train_Y[tt] = Train_Y[tt]- MeansResponse[1,dayIndex,month]
      tt <- (Test$Month == month  Test$Day == days_selected[dayIndex])
      if(length(grep(TRUE,tt))==0)
      {
        next
      }
```

```
      Test_Y[tt] = Test_Y[tt] - MeansResponse[1,dayIndex,month]
    }
}
Train <- sweep(Train,2, Means)
Train <-Train
Test <- sweep(Test,2, Means)
Test <-Test
Train$Year <- NULL
Train$Month <- NULL # month information is included in Day column.
Test$Year <- NULL
Test$Month <- NULL
rm(tt)
##############################################
##############2. Linear regression to get epsilon ################
##############################################
lmode <- lm(Train_Y ., Train[,2:dim(Train)[2]])
lmode$coefficients[grep(TRUE, is.na(lmode$coefficients))] <- 0
Residue <- lmode$residuals
##############################################
###########3. Get updated beta iteratively until converge#########
##############################################
max_iteration <- 20
iteration <- 1
####Get initial results from linear regression results.####
last_betas <- lmode$coefficients #keep track of betas
updated_betas <- mat.or.vec(length(last_betas),1)
E_Inv <- mat.or.vec(length(last_betas),length(last_betas))
E_Inv_Last <- mat.or.vec(length(last_betas),length(last_betas))
R2 <- mat.or.vec(1, max_iteration+1)
R2_Test <- mat.or.vec(1, max_iteration+1)
R2[,iteration] <- (t(Residue) %*% Residue/(length(Residue)-1))^0.5
predict_Y <- (last_betas[1] + matrix(unlist(Test[,2:dim(Test)[2]]),dim(Test)[1])
ResidueTest <- t(Test_Y-predict_Y)
R2_Test[,iteration] <- (ResidueTest %*% t(ResidueTest)/(length(ResidueTest)-1))^0.5
BETA_RECORDS <- mat.or.vec(length(last_betas),max_iteration)
BETA_RECORDS[,1] <- last_betas
TARGET_RECORDS <- mat.or.vec(1,max_iteration)
TARGET_RECORDS[,1]<- 1
```

```
OPTIMIZE_RESULTS <- mat.or.vec(1, max_iteration+1)
####Target function for Lasso####
lamda <- 10 * 1/(sqrt(dim(Train)[1]) * log(dim(Train)[1])) # 10 is a constant factor. We can try other values
fr <- function(lastbeta) ## Laso target function
{
  n <- dim(Train)[1]
  index <- rep(FALSE, length(lastbeta))
  for(i in 1: length(lastbeta))

    index[i] <- (index[i] — abs(lastbeta[i])<0.0001)

  lastbeta[index] <- 0;
  tt <- t(Train_Y- (lastbeta[1] + matrix(unlist(Train[,2:dim(Train)[2]]),dim(Train)[1])%*% lastbeta[2:length(lastbeta)]))
  s <- tt %*% E_Inv
  s - s %*% t(tt)
  abs(s/n) + lamda*sum(abs(lastbeta))
}
iteration = 1;
while(max(abs(updated_betas-last_betas)) > 0.001 & iteration < max_iteration)
{
  print("Iteration:")
  print(iteration)
  fileConn <- file(paste("Iteration",session,".txt",sep="")) #save iteration to disk
  writeLines(as.character(iteration), fileConn)
  close(fileConn)
  if(iteration>1)
  {
    last_betas <- updated_betas
  }
  #######
  ##3.2##covariance matrix w.r.t. time
  #######
  E <- mat.or.vec(dim(Train)[1],dim(Train)[1])
  U <- Residue
  U_Mean <- mean(U)
  for (j in 1:length(U))
  {
    for ( k in j:length(U))
```

```
  {
    #gamma_hat[h]
    E[j,k] <- E[j,k] + 1/length(U)*sum((U[1:(length(U)-k+1)]-U_Mean)*(U[k:length(U)]-U_Mean))
    E[k,j] <- E[j,k]
  }
}
#There is some problem in calculating inverse matrix using R. Using Matlab instead.
#E_Inv <- solve(E)
while(file.exists("MatlabInUse.txt"))
{
  Sys.sleep(30)
  next
}
fileConn <- file("MatlabInUse.txt") #take over matlab connection
writeLines(as.character(session), fileConn)
close(fileConn)
####Open Matlab connection####
Matlab$startServer()
matlab <- Matlab()
isOpen <- open(matlab)
if(!isOpen)
{
  Sys.sleep(30)
}
filename <- paste(tempfile(), ".mat", sep="")
dir.create(paste(Date_Path, "/CovarianceMatrix",session,"/", sep=""));
filename2 <- paste(Date_Path, "/CovarianceMatrix",session,"/",iteration ,".mat", sep="")
writeMat(filename, E=E)
evaluate(matlab, paste("load'",filename, "';",sep=""))
evaluate(matlab,"EInv = inv(E);")
evaluate(matlab, paste("save'",filename2,"'", "EInv",sep=""))
E_Inv <- readMat(filename2)
E_Inv <- unlist(E_Inv)
E_Inv <- matrix(E_Inv, ncol = length(E_Inv)^0.5)
unlink(filename)
evaluate(matlab,"exit;")
close(matlab)
Sys.sleep(5)
```

```
  file.remove("MatlabInUse.txt");
  #######
  ##3.4##Lasso to get updated betas
  #######
  optResult <- optim(last_betas, fr)
  updated_betas - optResult$par
  OPTIMIZE_RESULTS[iteration] - optResult$value
  #######
  ##3.5##update residue (Used to calculate covariance matrics)
  #######
  Residue <- Train_Y-(updated_betas[1] + matrix(unlist(Train[,2:dim(Train)[2]]),dim(Train)[1])% * %
          updated_betas[2:length(updated_betas)])
  iteration <- iteration+1
  BETA_RECORDS[,iteration] <- updated_betas
  R2[,iteration] <- (t(Residue) % * % Residue/(length(Residue)-1))^0.5
  predict_Y <- (updated_betas[1] + matrix(unlist(Test[,2:dim(Test)[2]]),dim(Test)[1])% * %
          updated_betas[2:length(updated_betas)]);
  ResidueTest <- Test_Y - predict_Y
  R2_Test[,iteration] <- (t(ResidueTest) % * % ResidueTest/(length(ResidueTest)-1))^0.5
  filename3 <- paste(Date_Path, "/CovarianceMatrix_",
  as.character(depths_elected), "/CurrentWorkspaceSave.mat", sep = "")
  save.image(filename3)
}
iteration <- iteration -1
#########################################
########4. Plot Results on both Train and Test data#######
#########################################
#######
##4.1##plot rmse
#######
rmse <- function(obs, pred) sqrt(mean((obs-pred)^2))
Train_RMSE <- mat.or.vec(1, iteration)
Test_RMSE <- mat.or.vec(1, iteration)
for (i in 1: iteration)
{
  Train_RMSE[i] <- rmse(Train_Y,(BETA_RECORDS[1,i] + matrix(unlist(Train[,2:dim(Train)[2]]),dim(Train)[1])%*%
          BETA_RECORDS[2:length(updated_betas),i]))
  Test_RMSE[i] <- rmse(Test_Y,(BETA_RECORDS[1,i] + matrix(unlist(Test[,2:dim(Test)[2]]),dim(Test)[1])%*%
```

```
        BETA_RECORDS[2:length(updated_betas),i]))
}
plot.new()
g_range <- range(0, Train_RMSE, Test_RMSE)
plot(t(Train_RMSE), type="o", col="blue", ylim=g_range, axes=TRUE, ann=FALSE)
lines(t(Test_RMSE), type="o", pch=22, lty=2, col="red")
title(main="rmse", col.main="red", font.main=4)
title(xlab="Iteration", col.lab=rgb(0,0.5,0))
title(ylab="rmse", col.lab=rgb(0,0.5,0))
legend(1, g_range[2], c("Train","Test"), cex=0.8, col=c("blue","red"), pch=21:22, lty=1:2);
#######
##4.2##plot beta convergence
#######
MAX_BETA_DIFF <- mat.or.vec(1, iteration)
MAX_BETA_DIFF[1] <- max(BETA_RECORDS[,1])
for (i in 2: iteration)
{
  MAX_BETA_DIFF[i] <- max(abs(BETA_RECORDS[,i]-BETA_RECORDS[,i-1]))
}
windows()
plot.new()
g_range <- range(0, MAX_BETA_DIFF)
plot(t(MAX_BETA_DIFF), type="o", col="blue", ylim=g_range, axes=TRUE, ann=FALSE)
title(main="Convergence trend of beta", col.main="red", font.main=4)
title(xlab="Iteration", col.lab=rgb(0,0.5,0))
title(ylab="
max diff with last iteration", col.lab=rgb(0,0.5,0))
```